Amendments to the Specification:

Please replace the paragraph beginning at page 8, line 29 with the following amended paragraph.

Referring now to FIG. 2, in accordance with the present invention, a system 30 that utilizes two-dimensional (2D) scene graph display commands 36a-36b provided by a software application 32 is shown. In this example, a button ~~32~~ 34 (including associated text) is represented by scene graph display commands 36a, 36b. One or more "create" commands 36b generate a scene graph data object corresponding to the button 34. The one or more "create" commands are received by hardware 38. The hardware 38 includes an API 38a, which, in conjunction with CPU 38b, transforms the one or more "create" commands to scene graph data corresponding to a button object, and stores the button object to a scene graph 38d. Once the button object is stored to the scene graph 38d, the button object can be invoked by a single render display command 36a. The button object can be invoked any number of times by respective "render " commands in order to display any number of respective button images on the monitor 38e. This is in contrast to the prior art approach of FIG. 1 in which multiple primitive "paint" commands are required to display each button.

Please replace the paragraph beginning at page 11, line 11 with the following amended paragraph.

Referring now to FIG. 3B, a graphics scene graph 100 associated with~~for~~ one of the buttons 80c of FIG. ~~3B~~ 3A includes as variety of components 102-116, of which components 102, 106, and 112 are group components. ~~The~~ a group component 112 is connected to area and text components 114, 116 respectively. The area and text components 114, 116 are associated with a single button image. The scene graph, which is comprised of scene graph data, underlies the component scene graph 70 of FIG. 3A and the graphics scene graph 100 of FIG. 3B. The scene graph is stored in the 3DGC, for example the 3DGC 38c of FIG. 2. Thus, FIGS. 3A and

3B illustrate diagrammatic representations of a scene graph used to provide the 2D image 50 of FIG. 3.

Please replace the paragraph beginning at page 10, line 1 with the following amended paragraph.

Referring briefly to FIG. 2A, tree diagram 40 is a diagrammatic representation of the scene graph 38d, and diagrammatically represents the data underlying the scene graph. A "button" group 40a includes a button portion 40b, a border portion 40c, and a text portion 40f. Arcs 40d, ~~and~~ 40e, and 40g couple the diagrammatic objects. While only a simple button object (including associated text) is shown, it should be realized that complex objects can similarly be diagrammatically represented and can similarly be associated with one or more "create" display commands.

Please replace the paragraph beginning at page 14, line 18 with the following amended paragraph.

Referring now to FIG. 6, an exemplary ATC graphical display 300 in accordance with the present invention is generated using a 2D scene graph and associated 2D scene graph display commands. The exemplary ATC graphical display 300 includes a variety of desktop images, including a menu having buttons with associated text, of which a button 302 is but one example, aircraft images 304a-~~304i~~304e, text 306 associated with the aircraft images, runway images 308 indicating an airport, and window images 310a-310e which can be zoomed and/or panned.

Please replace the paragraph beginning at page 8, line 14 with the following amended paragraph.

Referring now to FIG. 1, an example of a prior art system 10 for placing a button 14 (including associated text) on a monitor 18c includes a software application portion 12, which characterizes the button in terms of primitive "paint" display commands 16a-16c. The software

application ~~14~~ 12 sends display commands 16a-16c to the hardware 18, e.g., to a low-level application programming interface (API) 18a. A central processing unit (CPU) 18b, draws or "paints" the button ~~12~~ 14 on the monitor 18c. In the example of FIG. 1, the button ~~12~~ 14 is painted on the monitor 18c using three primitive "paint" display commands 16a-16c (i.e., paint(rectangle), paint(text), and paint(line)). Using these three primitive "paint" display commands 16a-16c, the software application 12 is able to cause the hardware 18 to display an image of the desired button 14.